Machine Vision and Applications (1990) 3:87–106

Machine Vision and Applications

On Improving the Accuracy of the Hough Transform

Wayne Niblack and Dragutin Petkovic
IBM Almaden Research Center, 650 Harry Road, San Jose, California, USA

Abstract: The subject of this paper is very high precision parameter estimation using the Hough transform. We identify various problems that adversely affect the accuracy of the Hough transform and propose a new, high accuracy method that consists of smoothing the Hough array $H(\rho, \theta)$ prior to finding its peak location and interpolating about this peak to find a final sub-bucket peak. We also investigate the effect of the quantizations $\Delta\rho$ and $\Delta\theta$ of $H(\rho, \theta)$ on the final accuracy.

We consider in detail the case of finding the parameters of a straight line. Using extensive simulation and a number of experiments on calibrated targets, we compare the accuracy of the method with results from the standard Hough transform method of taking the quantized peak coordinates, with results from taking the centroid about the peak, and with results from least squares fitting. The largest set of simulations cover a range of line lengths and Gaussian zero-mean noise distributions. This noise model is ideally suited to the least squares method, and yet the results from the method compare favorably. Compared to the centroid or to standard Hough estimates, the results are significantly better—for the standard Hough estimates by a factor of 3 to 10. In addition, the simulations show that as $\Delta\rho$ and $\Delta\theta$ are increased (i.e., made coarser), the sub-bucket interpolation maintains a high level of accuracy. Experiments using real images are also described, and in these the new method has errors smaller by a factor of 3 or more compared to the standard Hough estimates.

Key Words: high precision measurement, subpixel accuracy, straight-line detection, object location, Hough transform

1 Introduction

The Hough transform (Hough 1962; Duda and Hart 1972; Ballard and Brown 1982) provides a technique for deriving the values of parameters of a model given a set of points that includes instances of the model. Common uses include finding the parameters of a line or circle given a set of points, but it may be extended to other parameterized objects such as parabolas, ellipses, and so on, and it has been generalized to arbitrary shapes (Ballard and Brown 1982). A recent and thorough survey is given in Illingworth and Kittler (1988). It has been widely discussed in the literature, including theoretical aspects (Cohen and Toussaint 1977; Shapiro 1978; Sklansky 1978; Shapiro and Iannino 1979; Van Veen and Groen 1981; Brown 1983; Maitre 1986; Hunt et al. 1988; Srihari and Govindaraju 1989), efficient software implementations (Li et al. 1986; Illingworth and Kittler 1987), computation by special VSLI architectures (Baringer et al. 1987; Hinkle et al. 1987; Hanahara et al. 1988; Rhodes et al. 1988), computation by optical means (Eichman and Dong 1983; Gindi and Gmitro 1984; Ambs et al. 1986; Steier and Short 1986), and industrial applications (Li 1983; Dyer 1983). Although much has been published, practical questions on its accuracy and performance remain open. The objectives of this paper are to describe methods that achieve very high accuracy in parameter estimates using the Hough transform. We consider the case of finding the parameters of a straight line although the techniques apply to any use of the Hough transform to compute feature values and, in fact, apply to the generalized Hough transform. The improved accuracy is compared with that obtained from other methods, specifically the standard Hough technique of taking the absolute peak, with taking the centroid about the peak of the Hough array, and with least squares fitting. To start, several problems that adversely affect the accuracy of the Hough transform are described. We then investigate methods to overcome the problems that are based on preprocessing and interpolation in Hough space. To verify the methods, we use extensive simulation (various noise levels, line lengths, orientation, Hough space

Address reprint requests to: Wayne Niblack, IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, USA.

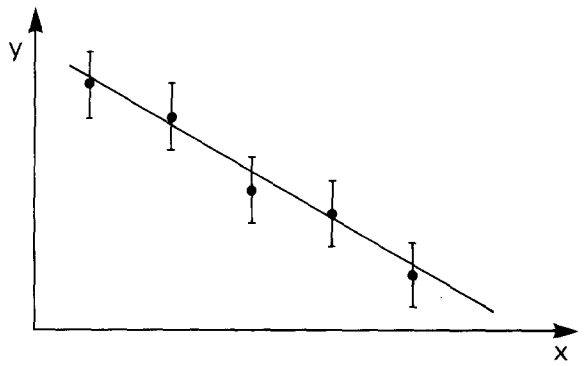

Figure 1. The noise model for $45° \leq \theta \leq 90°$. The vertical bars on each point represent, for example, a one sigma range about the point.

quantizations, etc.) and a number of experiments on real images of calibrated targets. The main emphasis of the paper is on the simulations because using real images brings in the problems of digital edge detection and optical and electronic distortion. For the experiments with real images we are mainly concerned with properly combining the edgel estimates to provide high precision line estimate, rather than high precision edgel detection. Clearly, improved edgel detection will increase the accuracy of our methods, but we do not address this and in our experiments we use straightforward edge detectors. Also, our work is geared toward machine vision, where knowledge of approximate line parameters (e.g., position, orientation, noise) is available. It is easily extended to less restricted domains, where a low-resolution Hough transform can be used to initially identify approximate line parameters to provide a smaller search region, such as that in Illingworth and Kittler (1987).

Our noise model for the points on a given line is similar to the one in Gordon and Seering (1986, 1988): we assume noise in only one of the coordinates. For lines whose normal make an angle θ with respect to the positive x-axis in the range [45°, 90°], the noise is assumed in y so that the method for deriving the edgels can be considered to have a deterministic x and noisy y. See Figure 1. For angles in the range [0°, 45°] we flip the problem and use a deterministic y and noisy x. This choice of deterministic x (or y) matches many computational methods of determining (x, y) edgel coordinates [see, e.g., Canny (1986) and Young (1986)]. In most of our simulations we used Gaussian zero-mean noise, but we also include tests with zero-mean uniform noise and with non-zero-mean noise.

We compare our results with results from least squares fitting. Assuming one line and zero-mean Gaussian noise, least squares gives the best estimate in the sense of maximum likelihood, and we show that our methods compare favorably with least squares even in the case to which the latter is ideally suited. In other cases, for example, multiple lines or lines with pattern or biased noise, standard least squares is not appropriate whereas the Hough transform may still be used.

Section 2 contains general introductory remarks and examples comparing the Hough transform and least squares for line parameter estimation. Section 3 reviews the Hough transform and previous work, and section 4 describes the new method for estimating the line parameters from the Hough array. Section 5 presents results from simulations using the methods, and section 6 has results from real images. Section 7 shows that the methods correspond to particular image domain filters, and section 8 contains conclusions.

2 Examples of Line Fitting Using Least Squares and the Hough Transform

The problem we are addressing is the one in which a set of points $\{(x_i, y_i), i = 1, \ldots, n\}$ is given, and we are to determine the parameters of instances of a known geometric model suggested by the points. We will consider the case where the model is of a straight line. The points are assumed to contain noise. Otherwise the problem is trivial. The noise can include "background noise," for example, uniformly distributed over the plane, or "pattern noise," by which we mean non-zero-mean noise such as from points along another line or curve, from a defect such as a "mousebite" along the line, and so on. Examples are shown in the figures given later in this section.

There are two well-known methods for solving this problem. The most common is least squares fitting. Least squares is computationally efficient, parameter free, and provides highly accurate estimates of line parameters in many cases. In particular, if the noise can be modeled as Gaussian and zero-mean (in fact, the conditions are more general), least squares has the desirable property that it produces the maximum likelihood estimate of the line parameters. In many practical problems it is an excellent choice. In addition, the basic method has been extended to robust least squares methods (Holland et al. 1977; Huber 1981), and these can be applied to situations containing outliers, certain cases of background and pattern noise, and various other cases for which standard least squares produces poor results.

An alternative to least squares is the Hough

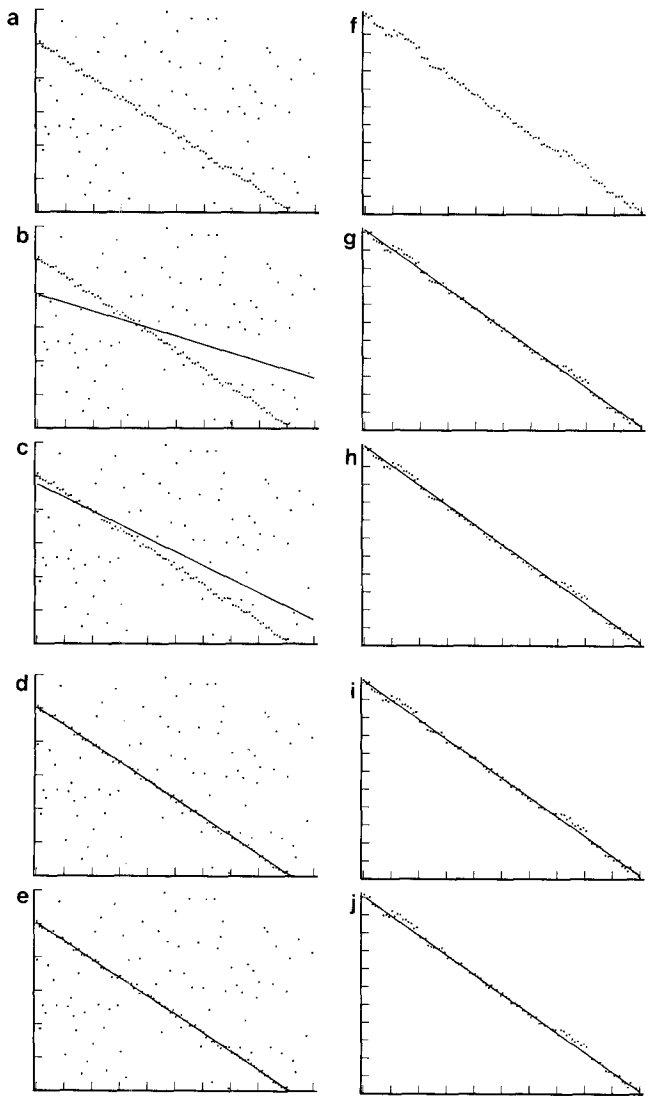

Figure 2. Left column, top to bottom, points and fits from a noisy line with background noise. (a) original points; (b) least squares fit minimizing y distance; (c) least squares fit minimizing perpendicular distance; (d) robust least squares; (e) Hough transform. Right column, top to bottom, points and fits from a noisy line with pattern noise. (f) through (j) corresponding to (a) through (e).

transform. The principal advantage of the Hough transform is its strong immunity to noise. Secondary advantages are the natural way a priori information on the approximate line parameters can be included, and for some cases its ability to handle multiple lines. Not often discussed is the accuracy of the computed parameters, which is the subject of this paper.

By way of introduction we give several examples. Figure 2a shows a set of noisy points from a line, together with background noise. The fit from a standard least squares, minimizing the sum of the squares of the errors in the y-coordinate, is shown in Figure 2b. Results from fitting to minimize the sum of the squares of the orthogonal distances to the line are shown in Figure 2c. Clearly, neither of these is satisfactory. Results from a robust least squares and from the Hough transform are shown in Figure 2d and 2e, and both are good. The same five figures are shown for the case of noisy points with sections of non-zero-mean pattern noise, Figure 2f through 2j. (The non-zero-mean noise segments are most clearly seen in the robust least squares and Hough transform fits as two sections of points slightly above the fitted line.) Also shown are cases with two lines. The first case has two nearly parallel intersecting lines (Figures 3a through 3e); the other case has two close parallel lines (Figures 3f through 3j. These are cases when even robust least squares does not uniquely identify the individual line. For all plots the robust least squares fits were done with the GRAFSTAT statistical and plotting package (GRAFSTAT User's Guide), which has a variety of robust regression techniques. We used the Beaton–Tukey option.

The examples give some idea of the type of problems where the various techniques are suitable. No technique can be directly applied in all cases, and each application must be evaluated and an appropriate technique selected.

3 Review of the Hough Transform and Previous Work

The parameters used to describe a line may be the slope and y-intercept. An alternative set of parameters is ρ and θ in the normal form (Duda and Hart 1972)

$$\rho = x \cos \theta + y \sin \theta \tag{1}$$

where ρ is the perpendicular distance of the line to the origin, and θ is the angle between a normal to the line and the positive x-axis. The normal form has several advantages over the slope-intercept form, related to the fact that ρ and θ vary uniformly as the line orientation and position change, and neither goes to ∞ as the line becomes horizontal or vertical. When using the Hough transform, input is a set of coordinates $\{(x_i, y_i), i = 1, \ldots, n\}$ of typically noisy points and output is the parameters ρ and θ describing a line through (or near) (some maximal subset of) the points.

The essential idea of the Hough transform is illustrated in Figure 4. A point (x_i, y_i) in Cartesian coordinates is mapped to all points in the ρ-θ parameter space that specify a possible line through the

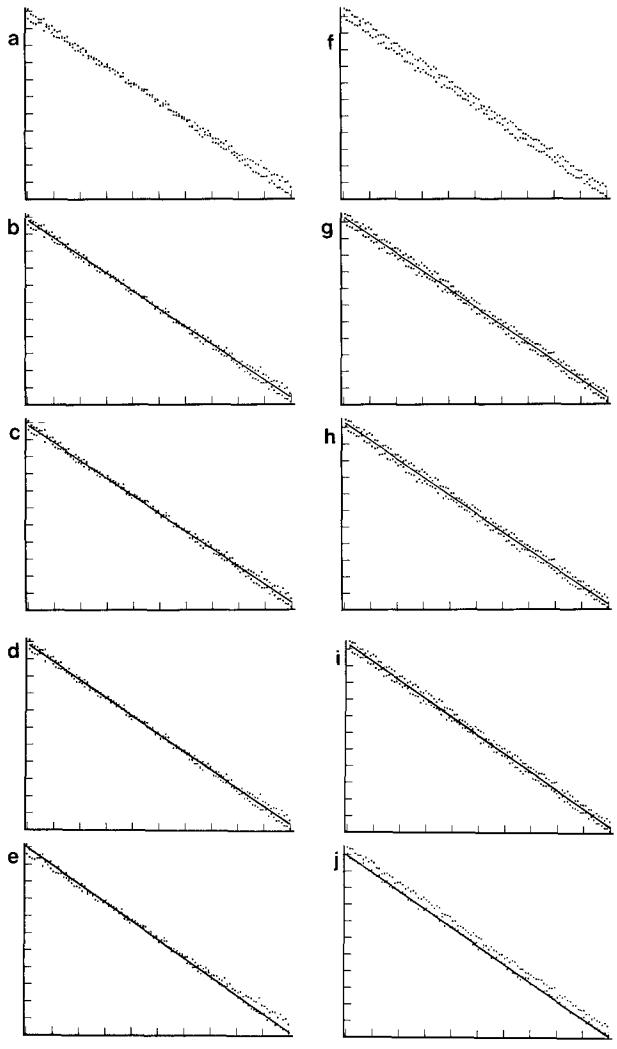

Figure 3. Left column, top to bottom, points and fits from two noisy nearly parallel lines. (a) original points; (b) least squares fit minimizing y distance; (c) least squares fit minimizing perpendicular distance; (d) robust least squares; (e) Hough transform. Right column, top to bottom, points and fits from two noisy parallel lines. (f) through (j) corresponding to (a) through (e).

point. From Eq (1) it is easy to show that this set traces a sinusoid in parameter space. Points on a particular line will all map to sinusoids that intersect in a common point and the (ρ, θ) of that intersection point gives the parameters of the line. To compute the parameters ρ and θ given a set (x_i, y_i), parameter space is quantized to $\mathbf{P} \times \Theta$, where ρ extends over a range $\mathbf{P} = \{\rho_0, \rho_1, \rho_2, \ldots, \rho_n\} = \{\rho_k | \rho_k = \rho_0 + k\,\Delta\,\rho\}$ and θ extends over $\Theta = \{\theta_0, \theta_1, \theta_2, \ldots, \theta_m\} = \{\theta_l | \theta_l = \theta_0 + l\,\Delta\,\theta\}$. An accumulator array $H(\rho, \theta)$ is defined on $\mathbf{P} \times \Theta$. Each point (x_i, y_i) is mapped to a set of locations in $\mathbf{P} \times \Theta$ by sampling θ (to the set Θ), using Eq (1) to compute ρ and quantizing this to the closest ρ in $\mathbf{P}$. In this way (x_i, y_i) maps to a sampled, quantized sinusoid in $\mathbf{P} \times \Theta$ and each accumulator in $H(\rho, \theta)$ along the sinusoid is incremented. When $H(\rho, \theta)$ has been filled in by all points, the (ρ, θ) location with the highest count is taken to indicate the parameters ρ and θ of the line that best explains the points. This is the most common method described in the literature, although many variations have been suggested. Of particular interest to us is that, due both to noise in the (x_i, y_i) and to the quantization of $\mathbf{P}$ and Θ, the sampled, quantized sinusoids do not in general intersect precisely at a common point, and some modification to the method is necessary if high accuracy in parameter estimation is desired.

Using the method as described above, the following three factors adversely affect the accuracy obtained. First, when accumulating $H(\rho, \theta)$, given a point (x, y), the bucket to increment in a particular θ column is computed by rounding ρ from Eq. (1). The fractional ρ information is lost and all subsequent computations are done on rounded ρ values. Another way of looking at this is that a given cell in $H(\rho, \theta)$ corresponds to a swath, and not a line, in the image. The swath is at angle θ and is centered at ρ with width $\Delta\rho$. ρ information finer than the swath width is lost. Second, after the accumulation is completed, an error may occur when selecting the cell in $H(\rho, \theta)$ with the maximum value if no consideration is made for the spreading of the peak in the ρ direction. This spreading, which was presented in Van Veen and Groen (1981), is discussed in detail below. The implicit assumption in the standard Hough approach is that all points from a line will fall in one bucket, but this is not necessarily true even in the noise-free case unless the θ of the line happens to be one of the θ's in the set Θ. And in the presence of even a small amount of noise, the maximum of $H(\rho, \theta)$ can occur at other than the (ρ, θ) in $\mathbf{P} \times \Theta$ closest to the (ρ, θ) of the line. Third, by taking the line parameters (ρ, θ) to be the coordinates of the peak, the quantization in ρ and θ is given by $\Delta\rho$ and $\Delta\theta$, limiting the accuracy. These are the three main factors we will examine. Due to these factors, the accuracy of the parameters computed by the standard Hough transform is often inferior to those computed by reasonable least squares techniques. (Reasonable means, e.g., that outliers are removed.)

A relevant comment is that the original patent for the Hough transform was not directly concerned with accuracy, but simply the recognition of line segments. The patent was motivated by analysis of bubble chamber photographs of subatomic particle tracks, which contain a large amount of noise, and

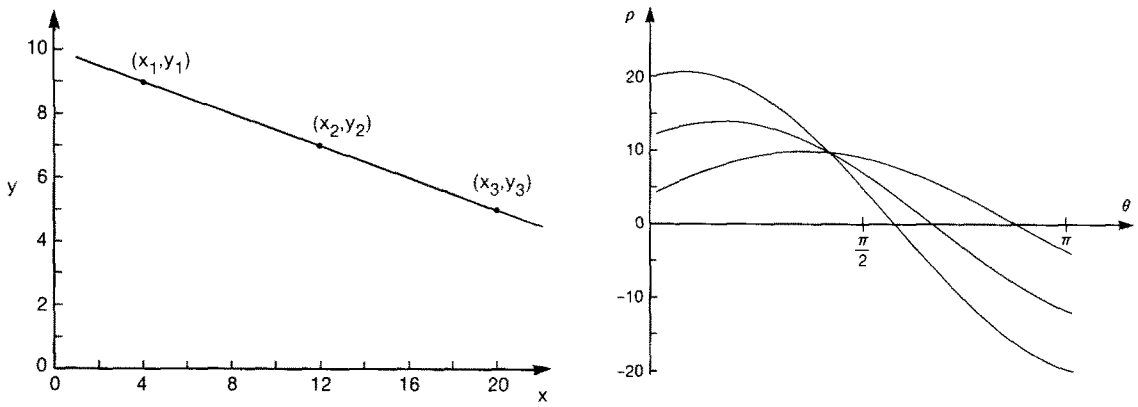

Figure 4. The Hough transform for points (x_i, y_i) and a (ρ, θ) parameter space.

proposed specific circuitry of amplifiers, delays, signal generators, and so on to perform what we now call the Hough transform in an analog form. It is as the technique has been implemented digitally, and applied in various applications such as machine vision and robot guidance, that the issue of accuracy has become important.

Since it will be useful later, we mention here that the Hough transform is closely related to the Radon transform. [See, e.g., Deans (1983)] The Radon transform of a function $f(x, y)$ is defined as

$$R(\rho, \theta) = \int_{L(\rho,\theta)} f(x, y)\, ds = \int_{x,y} f(x, y)\delta(\rho - x\cos\theta - y\sin\theta)\, dx\, dy \quad (2)$$

where $L(\rho, \theta)$ is a line at angle θ and distance ρ from the origin. Thus, the Hough transform is the (discrete) Radon transform of the edgel image (Deans 1981). For a (binary) edgel image a column of θ of the Radon or Hough transform is a projection of the image at the given θ. Also, for any θ

$$\sum_{\text{all possible } \rho} H(\rho, \theta) = \text{constant} = \text{the total number of edgels} \quad (3)$$

As mentioned earlier, noise in the (x_i, y_i) and the quantizations of $\mathbf{P}$ and Θ cause a spreading of the peak in $H(\rho, \theta)$. Other factors also influence the peak, and numerous authors have considered issues related to the accuracy of the Hough transform. In Cohen and Toussaint (1977) the authors model the combined effect of background noise and the geometry and extent of a circular retina (for our case, the image array) on the values accumulated in $H(\rho, \theta)$ and suggest a useful method for "standardizing" the counts in $H(\rho, \theta)$ by normalizing by the expected mean and standard deviation at each cell to improve the peak finding. Similarly, the analysis in Maitre (1986) considers the detection of straight lines in rectangular retinas and derives a signal-to-noise ratio based on the line length, the spread of points about the true line, and the number of line points and noise points. In Hunt et al. (1988), detecting lines using the Hough transform is done using a likelihood ratio of the two hypotheses—(1) the image contains some line and (2) the image does not contain a line—and uses a statistical approach that takes into account the length of the line at (ρ, θ) (i.e., the effect of the geometry of the retina) and an a priori probability distribution of lines in the image. Brown (1983) describes inherent noise and bias in $H(\rho, \theta)$ due to multiple instances of features in an image and describes the complementary Hough transform, which attempts to reduce interference between different peaks, especially for radially symmetric objects. Ballard (1987) describes a form of locally linear interpolation on the Hough array to achieve high accuracies when moving from low-level continuous data to high-level discrete data in the context of neural models for perception. This is one of the few papers specifically proposing interpolation on $H(\rho, \theta)$. Recently, an anti-aliasing method, treating Hough space $H(\rho, \theta)$ as a sampled signal and applying a Nyquist-like criteria to recovery information, provides a digital signal processing approach to analyzing the accuracy of the Hough transform (Kiryati and Bruckstein 1988).

Sklansky (1978) proposed a geometric construction allowing the object parameters to be represented in x–y space, and in Shapiro (1978) the author uses this construction to propose bounds for the spread in the computed parameters (the "solution region") as a function of the bounds on the errors in the (x_i, y_i). In contrast to his work, we use a different noise model (Gaussian versus absolute

bounds) and different approach (analytic versus geometric). We propose smoothing and interpolation in Hough space to achieve high accuracy, whereas he determines the quantizations (i.e., accumulator cell size) to be on the order of the spread that can occur from points from a single line and then takes those quantizations as the achievable accuracy. One result is that our algorithm uses all the points in a line, whereas Shapiro's solution region depends only on the line end points.

Several techniques for improving the Hough transform accuracy use gradient information, which is often available with each edgel. Using gradient direction (Kimme et al. 1975; Ballard 1981; Sheinvald 1989) a constrained Hough transform can be computed. This typically saves significantly on the computation cost. In addition, it improves the accuracy by keeping both noise points and points from intersecting lines (which are near the point of intersection) from influencing the peak location. Gradient magnitude may also be used (Ballard 1981; Van Veen and Groen 1981; Davies 1987). The magnitude is used to determine the amount by which $H(\rho, \theta)$ is incremented. This allows strong edge points, assumed to be the most accurate, to contribute heavily to the result. These are valuable methods, primarily affecting the accumulation of $H(\rho, \theta)$. They can be used together with the techniques we will describe, which address the accumulation and, in particular, interpolation in $H(\rho, \theta)$ to achieve higher, sub-bucket accuracies.

Other approaches, such as the fast Hough transform (FHT) (Li et al. 1986) and the adaptive Hough transform (AHT) (Illingworth and Kittler 1987) , are coarse-to-fine iterations that can be run until a prespecified resolution in the parameters is obtained. These methods are efficient ways of accumulating and finding the absolute peak in $H(\rho, \theta)$, and if the centroid weighting of the AHT is considered, finding a sub-bucket peak. The main features of our approach are a smoothing of the accumulated array and an interpolation to a sub-bucket peak location. It does not search for the absolute integer peak, but rather, for an interpolated, smoothed peak that we claim is more accurate.

In a useful paper Van Veen and Groen (1981) consider the influence of the quantizations $\Delta\rho$ and $\Delta\theta$, the quantization of the image, and the width of the line segment on the Hough transform. They analyze ideal (thin) and thick lines, but not noisy lines. They derive formulas specifying peak spread in Hough space for a given line length and quantization and suggest searching for peaks in $H(\rho, \theta)$ over a sliding window. We use their results and extend them for the case of noisy lines and to sub-bucket accuracy. Gordon and Seering (1986, 1988), although they consider least squares fitting and not the Hough transform, investigate the accuracy of finding straight lines in digital images. Their papers contain a good balance of theory, simulation, and experiments. We will evaluate the results we obtained from the Hough transform using a similar methodology.

Many other references and issues can be cited. Our emphasis is on the following questions:

1. How significant are the three error sources listed previously and how can the compensation for them be made?
2. What is the accuracy achievable using the Hough transform with such compensations? How does it compare with that from other techniques, specifically least squares?
3. How should $\Delta\rho$ and $\Delta\theta$ be chosen and what effect do they have on the accuracy of the results? Can interpolation be used to achieve accuracies better than $\Delta\rho$ and $\Delta\theta$, and if so, what type of interpolation?

4 High Accuracy Estimation of Line Parameters from Hough Transform

We begin by considering how the peak spreads in $H(\rho, \theta)$ and then give methods for estimating the true peak location in the presence of this spreading.

4.1 The Spreading of the Peak in $H(\rho, \theta)$

Van Veen and Groen (1981) show that for a given $\Delta\theta$ and an ideal thin line of length L the worst-case spread of the peak $H(\rho, \theta)$ along the ρ coordinate is

$$s_\rho = L \sin(\Delta\theta/2) \tag{4}$$

In terms of buckets of ρ the worst-case spread is

$$s_\rho = \left[\frac{L \sin(\Delta\theta/2)}{\Delta\rho}\right] + 2 \tag{5}$$

where $[z]$ indicates the largest integer strictly smaller than z. If noise is present that spreads the estimates of the points from the line a perpendicular distance up to b away from the line, or in the case of a thick line (e.g., after thresholding the gradient) whose width is $2b$, then the worst-case spreading is

$$s_\rho = L \sin(\Delta\theta/2) + 2b \cos(\Delta\theta/2) \tag{6}$$

and in $\Delta\rho$ units

$$s_\rho = \left[\frac{L \sin(\Delta\theta/2) + 2b \cos(\Delta\theta/2)}{\Delta\rho}\right] + 2 \tag{7}$$

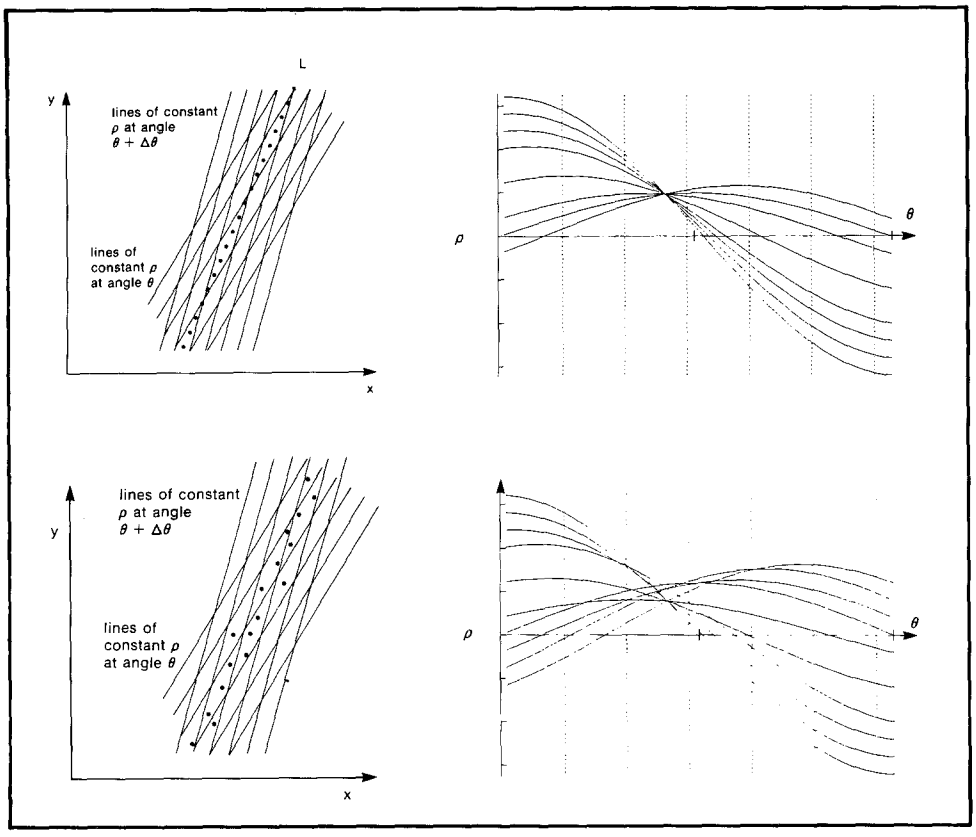

Figure 5. The spreading of the peak in ρ due to θ sampling and to noise.

The important conclusion from these formulas is that there is not necessarily a single "peak bucket" in $H(\rho, \theta)$ corresponding to the points along a given line. This, however, is a common assumption.

The two cases (with and without noise) are illustrated in feature space and in parameter space in Figure 5. These are easily visualized using a projection (or Radon transform) of the line edgels. Larger values of $\Delta\rho$ reduce the spreading in units of $\Delta\rho$ but result in loss of accuracy. Smaller values increase the spreading and also the computational cost and sensitivity to noise. Similarly, larger values of $\Delta\theta$ increase the worst-case spreading and also result in loss of accuracy.

The numeric example in Figure 6 shows a case where the peak spreads across several values of ρ. It was computed from a set of 200 noisy points from the line with $\rho = 25$ and $\theta = 30$. The noise was Gaussian with zero mean, $\sigma_{\text{Noise}} = 1$, and for the Hough array, $\Delta\rho = 0.1$ and $\Delta\theta = 0.1$. The example is also a case in which the absolute maximum (the value 22 at $\rho = 24.9$, $\theta = 29.9$) does not give the correct ρ and θ estimate.

The preceding paragraphs describe the spreading in ρ. Now consider the spreading in θ. Figure 7 shows a noise-free line segment of length L and the

	29.5					30.0					30.5
	7	6	6	9	5	4	10	5	7	12	6
24.50-	3	6	10	11	8	8	5	12	5	5	4
	6	10	5	12	11	13	15	10	5	4	7
	3	4	6	9	12	12	11	5	9	7	6
	5	4	9	7	14	14	10	13	8	6	7
	7	4	8	13	22	17	14	8	12	9	6
25.00-	2	8	8	16	11	17	14	19	12	10	5
	6	8	17	11	14	9	12	11	8	8	8
	7	13	10	12	6	17	14	7	13	4	5
	7	4	8	8	15	12	13	12	5	10	3
	5	12	3	7	12	13	11	11	7	8	6
25.50-	9	2	7	7	5	11	13	8	9	7	7
	5	3	6	9	14	10	11	5	9	1	10

Figure 6. A portion of $H(\rho, \theta)$ showing the peak spread and wrong solution if simple maximum is used. The corect solution is $\theta = 30$ and $\rho = 25$. The maximum occurs at $\theta = 29.9$ and $\rho = 24.9$.

swaths defined by the Hough transform for two successive values of θ. The line segment fits completely within a single swath for both θ's, so that the peak is spread in the θ direction. To have a unique peak, $\Delta\rho$ must be sufficiently small. The worst case for this type of ambiguity occurs when the θ of the line

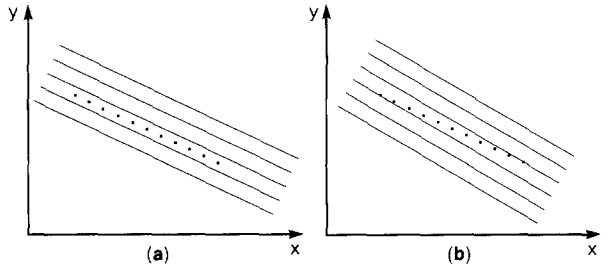

Figure 7. A case where all points fall in a single ρ bucket of different values of θ. This causes the peak to spread in the θ direction.

exactly matches one of the θ's in Θ. In this case we must have

$$\Delta\rho < L \sin(\Delta\theta) \tag{8}$$

to ensure that we can identify the correct θ, and this equation provides an approximate upper bound on the size of $\Delta\rho$. This is in case of no noise. Noise along the line makes the problem less severe by spreading the points over adjacent buckets, and the limit given by Eq. (8) can be relaxed.

4.2 Selecting the Quantizations $\Delta\rho$ and $\Delta\theta$ of $H(\rho, \theta)$

To compute the Hough transform, ranges of ρ and θ and the quantization $\Delta\rho$ and $\Delta\theta$ must be selected for $H(\rho, \theta)$. These parameters affect the memory size, processing times, and final accuracy of the computed parameters. As shown in Eq. (4) to (8), the values of $\Delta\rho$ and $\Delta\theta$ combine to influence the spreading of the peak so that they cannot be selected fully independently, which is a common practice.

Many implementations use fixed quantizations, such as $\Delta\rho = 1$ and $\Delta\theta = 1$, as well as a fixed range for ρ and θ. Hardware requirements may impose restrictions. For example, in Leavers and Boyce (1986), $H(\rho, \theta)$ is equal to an $n \times n$ frame buffer in which Hough transforms of $n \times n$ images are to be accumulated. This leads to $\Delta\rho = \sqrt{2}$ and $\Delta\theta = 180/n$. Other cases assume the accuracy of the computed values of ρ and θ are directly given by $\Delta\rho$ and $\Delta\theta$; for example, they assume the final ρ is accurate to within $\pm\Delta\rho/2$. And in cases where noise in the input points is considered some authors suggest selecting $\Delta\rho$ and $\Delta\theta$ to be comparable to the peak spread caused by the noise (Shapiro and Iannino 1979).

We will assume no hardware-based restrictions and that the quantizations can be freely selected. Computationally, a fine quantization plus a large range for the parameters requires a large array and much processing. Quantizing ρ to 0.1 over the range 0 to $(512 \times \sqrt{2})$, and quantizing θ to 0.1 from 0° to 180°, requires $724/0.1 \times 180/0.1 = 12.5M$ elements in $H(\rho, \theta)$, which is excessive. In industrial problems, a prior information such as mechanical fixture accuracy may be available and can be used initially to limit the ρ and θ ranges. This was the case for our tests where we knew the true values and therefore limited the array to a few units around the true solution. Another way to keep a manageable array size is a coarse/fine approach (Li et al. 1986; Illingworth and Kittler 1987), which iteratively decreases $\Delta\rho$ and $\Delta\theta$ as the peak is localized and the ranges of ρ and θ are reduced. The main point for us is that in either case the use of fine quantizations is computationally feasible if they are necessary.

We started by selecting fine (i.e., small) values of $\Delta\rho$ and $\Delta\theta$ in order to investigate the accuracies we could obtain. We use Eq. (6) to help ensure that the choices for $\Delta\rho$ and $\Delta\theta$ were compatible, given the noise characteristics and the length of line to be found. The intent was to have a spread near the peak in the ρ direct that was in the range 6–12. In this way, the form of the spread is preserved, which we need for our later interpolations. Also, we wanted to satisfy the maximum ρ requirement of Eq. (8). (In typical runs this requirement tends to give a smaller $\Delta\rho$ than Eq. (6) and hence a larger peak spread, but this causes no problems.) These equations provided approximate guidelines for the relation between $\Delta\rho$ and $\Delta\theta$, and we rounded them to convenient values. We ran with these values, accumulating $H(\rho, \theta)$ and interpolating to sub-bucket locations of $H(\rho, \theta)$ to find line parameter estimates with accuracies better than $\Delta\rho$ and $\Delta\theta$. In subsequent runs we used coarser values of $\Delta\rho$ and $\Delta\theta$ to determine if the sub-bucket interpolation can still produce the high accuracies, thus saving on array size and processing time.

Specifically, for the fine quantizations we started with the goal of obtaining an accuracy in θ of $\pm$ 0.1° and chose $\Delta\theta = 0.05°$. From Eq. (7), assuming a minimum line length $L = 100$ and $\sigma_{\text{noise}} = 1$, to get a peak spread of 6–12 $\Delta\rho$ buckets, we would have $\Delta\rho$ of approximately 0.5. But the maximum $\Delta\rho$ from Eq. (8) is 0.087. Because of the noise, we were safe in increasing this slightly, and the value we used was 0.1. Other runs with coarser quantization values used $\Delta\rho = 0.4$, $\Delta\theta = 0.25$; $\Delta\rho = 0.5$, $\Delta\theta = 0.5$; and $\Delta\rho = 1$, $\Delta\theta = 1$. In all cases we chose $\Delta\theta$ and then used Eq. (7) and (8) to guide the selection of $\Delta\rho$.

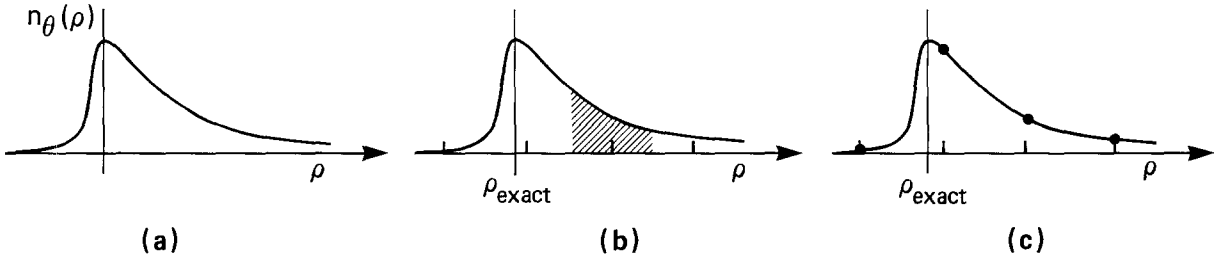

Figure 8. Methods of updating $H(\rho, \theta)$ given a noise model $n\theta(\rho)$.

4.3 Accumulating $H(\rho, \theta)$

Given $\Delta\rho$, $\Delta\theta$, and the input points, the next step is to accumulate $H(\rho, \theta)$. This can be done in various ways. The most straightforward is for each (x_i, y_i) and for each θ in Θ, compute ρ from Eq. (1), round it to the nearest ρ in $\mathbf{P}$, and increment the corresponding element of $H(\rho, \theta)$. This loses the fractional information in ρ.

To reduce this rounding error, the increment value (typically 1), can be distributed over a region in $H(\rho, \theta)$. This has been suggested by various authors (Shapiro 1978; Thrift and Dunn 1983; Galkowski and Galkowski 1986; Ballard 1987). One way is to distribute the increment over the nearest cells in a column (of constant θ) by, say, a linear distribution as follows. Let (x_i, y_i) be one of the points, let θ be a value in Θ, and let $\rho_{exact} = x_i \cos\theta + y_i \sin\theta$. Then if ρ_{low} and ρ_{high} are consecutive values of $\mathbf{P}$ such that $\rho_{low} \leq \rho_{exact} < \rho_{high}$, then increment $H(\rho_{low}, \theta)$ by an amount proportional to $\rho_{high} - \rho_{exact}$ and increment $H(\rho_{high}, \theta)$ by an amount proportional to $\rho_{exact} - \rho_{low}$. These two methods, rounding ρ_{exact} and incrementing the corresponding $H(\rho, \theta)$ bucket, and linearly distributing the increment to the buckets for ρ_{low} and ρ_{high}, are compared in the simulations given later.

Other increment methods are also possible. Given a noise model for the (x_i, y_i), the increment value can be distributed about its true sub-bucket location based on the model. As an example, let $n_\theta(\rho)$ be the noise in the ρ estimate at angle θ given point (x_i, y_i), Figure 8a. If ρ_{exact} is as above, then an exact updating of $H(\rho_j, \theta)$ can be done by incrementing with

$$\int_{\rho_j - \Delta\rho/2}^{\rho_j + \Delta\rho/2} n_\theta(\rho - \rho_{exact})\, d\rho \tag{9}$$

as shown in Figure 8b. This is expensive computationally. A less expensive but still precise method is to update with

$$n_\theta(\rho_j - \rho_{exact}) \tag{10}$$

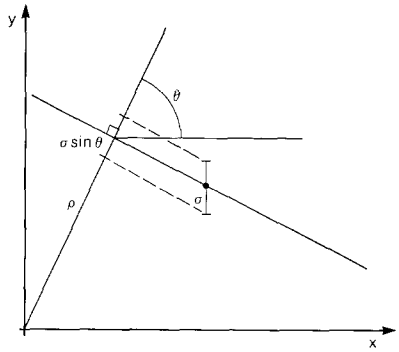

Figure 9. Noise in ρ resulting in noise from y_i at angle θ.

See Figure 8c. We discuss below [Eq. (11)] a related smoothing applied to $H(\rho, \theta)$ after all values have been accumulated.

4.4 Finding the Peak in $H(\rho, \theta)$

Assume now that we have accumulated $H(\rho, \theta)$ by some method. Instead of taking the coordinates (ρ_m, θ_m) of its maximum as the solution parameters, we want to achieve higher accuracy by compensating for the two remaining error sources: the peak spread and the quantizations $\Delta\rho$ and $\Delta\theta$ in the set $\mathbf{P} \times \Theta$. We model the spread in the peak based on a model of the noise in the points (x_i, y_i). This noise, which is assumed to be in one coordinate only (the y_i for θ in [45°, 90°]), is further assumed to be Gaussian with zero mean and variance σ^2_{Noise}. [See also Niblack and Petkovic (1986) where we also modeled the noise by a uniform distribution of unknown spread and found the spread by an iterative technique]. For a given θ the Gaussian distribution in a y_i induces a Gaussian distribution in the corresponding estimate of ρ with $\sigma_\rho = \sigma_{Noise} \sin\theta$, shown in Figure 9. As mentioned previously, ideally one would take each point (x_i, y_i) and for each θ in Θ compute the corresponding floating point ρ_{exact} as $\rho_{exact} = x_i \cos\theta + y_i \sin\theta$. Then for each ρ in a neighborhood of ρ_{exact} in $\mathbf{P}$ update the bucket $H(\rho, \theta)$ by an amount derived from the noise model. With

the Gaussian noise model the exact update factor [Eq. (9)] becomes

$$\frac{1}{\sqrt{2\pi}\sigma_\rho}\int_{\rho-\Delta\rho/2}^{\rho+\Delta\rho/2} e^{-(\rho'-\rho_{\text{exact}})/2\sigma_\rho^2}\,d\rho'$$

or, sampling at the bucket centers [Eq. (10)], it becomes

$$\frac{1}{\sqrt{2\pi}\sigma_\rho} e^{-(\rho' - \rho_{\text{exact}})/2\sigma_\rho^2} \tag{11}$$

To reduce the computation, we accumulate the $H(\rho, \theta)$ array and perform the "noise spreading" as a subsequent step. That is, we spread accumulated values instead of accumulating spread values. The spreading is identical to what would be done for each individual point, and is done by convolving each value with

$$\frac{1}{\sqrt{2\pi}\sigma_\rho} e^{-\rho^2/2\sigma_\rho^2}$$

The convolution is done as a one-dimensional spatial convolution in the ρ dimension along columns of θ. The value used for σ_ρ is an approximation that includes both the Gaussian noise coming from the noise in the y_i, and the non-Gaussian spreading of the peak due to the line length and quantization $\Delta\theta$. It is computed as

$$\sigma_\rho = \left[\frac{L\sin(\Delta\theta/2) + 2(3\sigma_{\text{Noise}})\cos(\Delta\theta/2)}{2}\right]\sin\theta \tag{12}$$

After the convolution the coordinates of the quantized peak (ρ_m, θ_m) are found by direct search. We were initially concerned that multiple maximums would occur and checked for these. None was detected in any of our simulations, due in part to the fact that each θ column is convolved with a different Gaussian (the sin θ term). And although it cannot be proved that only a single maximum will occur around the peak for a given line, our smoothing method should remove multiple peaks if the noise model and parameters are reasonably accurate.

Once the quantized peak is found, the coordinates of the final peak are found by a set of one-dimensional interpolations, first in the ρ direction along columns of θ and then in the θ direction. In each of the three θ columns θ_{m-1}, θ_m, and θ_{m+1} we fit a Gaussian to the eight points surrounding the maximum in the column. From the fitted Gaussians we have the location ρ_{H_m} and value H_m of the maximum for each column m. We now fit a Gaussian in the θ direction, using the points (θ_{m-1}, H_{m-1}), (θ_m, H_m) and (θ_{m+1}, H_{m+1}), and find its maximum, giving θ_{peak}. At θ_{peak}, which in general falls between two columns in $H(\rho, \theta)$, we linearly interpolate between the two surrounding values of ρ_{H_m} to get ρ_{peak}, giving the final interpolated result $(\rho_{\text{peak}}, \theta_{\text{peak}})$.

5 Simulation Results

We ran extensive simulations to evaluate the accuracy of the methods. The simulations compare the accuracy of four methods of computing ρ and θ: (1) using least squares, (2) smoothing $H(\rho, \theta)$ and interpolating to the final ρ and θ as described, (3) taking the centroid about the peak, and (4) taking the absolute peak. The centroid was included in the simulations since it has been suggested as a means of getting sub-bucket accuracy. For our tests, to compute the centroid, we took the weighted average of all cells in $H(\rho, \theta)$ around the peak whose value was greater than or equal to 80 percent of the peak. (We initially used 90 percent as the threshold, a value used in Illingworth and Kittler (1987), but this gave less accurate results. In general, the centroid method is sensitive to the threshold chosen.)

The main simulations used "nice" data, points from a single line with Gaussian zero-mean noise. Additional simulations considered various other cases: uniformly distributed noise, multiple noisy lines, and several types of pattern and non-zero-mean noise. The main simulations used

- $(\Delta\theta = 0.05, \Delta\rho = 0.1)$, $(\Delta\theta = 0.25, \Delta\rho = 0.4)$, and $(\Delta\theta = 0.5, \Delta\rho = 0.5)$, and $(\Delta\theta = 1, \Delta\rho = 1)$.
- Angles θ_{true} from 45° to 90°. Each angle was determined as $\theta_{\text{true}} = 45 + i + U(0, 0.5)$, $i = 0, 1, \ldots,$ 45. $U(0, 0.5)$ is uniformly distributed noise in the range ±0.5°. In this way we avoid any systematic error that would result from having only integer θ values.
- Line lengths L of 100, 200, 300, 400, and 500. We defined points (x_i, y_i) at each successive integer value of x and to keep the line length constant, the number of points per line varies as $L \sin\theta$.
- No noise and zero-mean Gaussian noise in the y_i with $\sigma_{\text{Noise}} = 0.5$ and $\sigma_{\text{Noise}} = 1.0$.
- y_i values rounded to integer and y_i kept as floating point.
- Two ways of accumulating the array $H(\rho, \theta)$; linear distribution of the increment value to the two surrounding ρ buckets, and incrementing (by 1) only the nearest ρ bucket as described in section 4.3 "Accumulating $H(\rho, \theta)$."

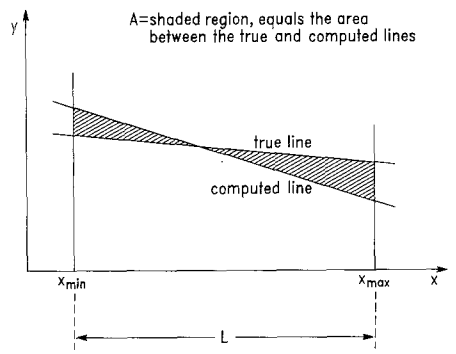

Figure 10. The error in a line was computed as A/L.

Initially three values of ρ_{true} were used: 15.00, 15.25, and 15.33. The last two were chosen to fall between buckets of ρ in $H(\rho, \theta)$. No significant differences were noted in the results for the three values, so the first and last were dropped and only the one value, $\rho = 15.25$ was used.

In each simulation (i.e., each set of parameters ρ_{true}, θ_{true}, L, $\Delta\rho$, $\Delta\theta$, rounded/floating point y_i, and with or without linear distribution), if $\sigma_{Noise} = 0$, we performed a single trial, computing estimates $\hat{\rho}_{edge}$ and $\hat{\theta}_{edge}$ from each of the four algorithms. If $\sigma_{Noise} \neq 0$, we performed 50 trials generating different random noise samples each trial. This produced 50 estimates $\hat{\rho}_{edge}$ and $\hat{\theta}_{edge}$. For each of these we computed the average error in the resulting lines as follows. Let x_{min} and x_{max} be the minimum and maximum of the input x_i. Then the error ε in a given line is A/L where A is the area between the true and computed line between x_{min} and x_{max}, and L is $x_{max} - x_{min}$, as shown in Figure 10. This removes the dependence of the error on the coordinate system origin.

For all runs the value of σ_ρ used to control the Gaussian convolution [Eq. (12)] was equal to σ_{Noise} for the cases using exact (floating point) y coordinates, and to $\sigma_{Noise} + 0.5$ when y was rounded to integer as an approximate compensation for the noise due to the rounding.

The simulation software is written in C. One set of simulations over all angles and all line lengths, with 50 noise samples of a fixed σ_{Noise}, took about nine hours of CPU time on an IBM 4381 (about 4.5 MIPS).

Selected graphical results from the simulations are shown in Figures 11 to 19. In all the graphs results marked with "○" are from least squares, with "+" from the $H(\rho, \theta)$ smoothing and interpolation, with "∇" from the centroid around the peak, and with "×" from the absolute peak.

Figure 11 has a set of plots of the average error in the computed lines for the four different $\Delta\rho$ and $\Delta\theta$ quantizations. The patterns shown in the figure are true of essentially all the cases we ran and demonstrate one of our main conclusions—that proper smoothing and interpolation can significantly improve the accuracy. Taking the absolute peak in the Hough array as the solution consistently gives the largest errors, the centroid is more accurate, and the smoothing and interpolation method is even more accurate. Least squares fitting, for this case of zero-mean noise, is the best. [Our least squares results agree with those reported in Gordon and Seering (1986, 1988).

The results from Figure 11 are for zero-mean noise with $\sigma_{Noise} = 1.0$, full precision y_i coordinates. Corresponding results for rounded y_i are shown in Figure 12. The differences in these two figures, as in other subsequent figures, are due to the algorithm differences and not to the random noise since the random noise generator was started with the same seed in the different cases. The most noticeable difference is the spikes that occur at various angles in the centroid and absolute peak methods. These will be discussed later. In other respects, particularly the accuracy of the smoothing and interpolation methods, the results are essentially unaffected by the rounding. A third set of results is shown in Figure 13, corresponding to Figure 11 but the zero-mean uniform noise of spread ±2. The y_i were exact. Here again, the smoothing and interpolation algorithm was the most accurate with one exception: For fine quantizations (Figure 13a) the smoothing and interpolation method became less accurate as the angle increased from 45° to 90°. In this case the peak of the Hough array is spread because the slope of the curves $\rho = x \cos\theta + y \sin\theta$ that intersect at the peak, given by $\partial\rho/\partial\theta = -x \sin\theta + y \cos\theta$, becomes small as θ approaches 90° in the case of small x (our data was centered around $x = 0$). At the same time, the fine quantizations of $\Delta\rho$ and $\Delta\theta$ and wide spread of uniform noise in the y_i give few counts in each bucket of $H(\rho, \theta)$ and hence a low signal-to-noise ratio. Several columns of $H(\rho, \theta)$ (i.e., several values of θ) all have similar values and the interpolation is fitting a Gaussian to an essentially constant value, giving lower accuracy results. Thus, coarser quantization in $\Delta\rho$ and $\Delta\theta$ actually improved the results. In all the previous cases we used linear distribution of the increment.

The results of Figure 11 are summarized in Figure 14, which shows the effect of the quantizations $\Delta\rho$ and $\Delta\theta$ on the computed accuracy. This is our second main result, and shows that fine values of $\Delta\rho$ and $\Delta\theta$ are not necessary, but that the interpolation can be used to achieve very high accuracies when

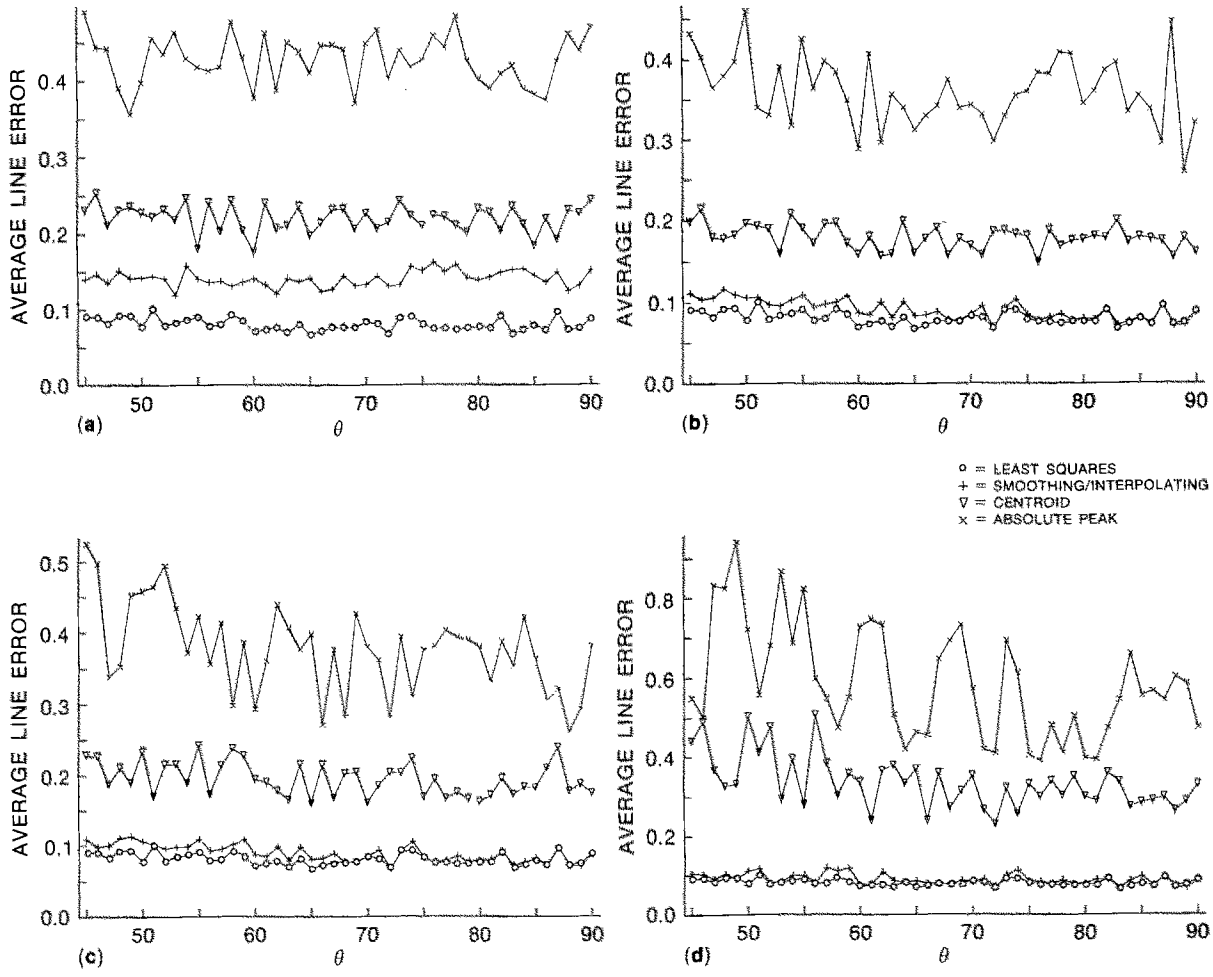

Figure 11. Average line errors for four quantizations of $\Delta\rho$ and $\Delta\theta$. (a) $\Delta\rho = 0.1$, $\Delta\theta = 0.05$; (b) $\Delta\rho = 0.4$, $\Delta\theta = 0.25$; (c) $\Delta\rho = 0.5$, $\Delta\theta = 0.5$; (d) $\Delta\rho = 1$, $\Delta\theta = 1$. Statistics were computed over 50 trials at each angle, updating $H(\rho, \theta)$ by linear distribution, $\sigma_{\text{Noise}} = 1.0$, full precision y_i, $L = 300$, $\rho = 15.25$.

using "coarse" $\Delta\rho$ and $\Delta\theta$. The sub-bucket interpolation is valid over a range of quantizations, and accuracies much higher than the bucket sizes can be obtained.

Figure 15 has plots summarizing results over different line lengths. Plots are given as a function of line length for average line error over all angles, 50 trials per angle. These are all from runs with zero-mean noise with $\sigma_{\text{Noise}} = 1.0$ in y_i, $\Delta\rho = 0.5$, $\Delta\theta = 0.5$, $H(\rho, \theta)$ increment by linear distribution, and y_i kept as floating point. The general pattern shown in these plots was true for other runs with noisy data using other values of $\Delta\rho$ and $\Delta\theta$, with exact versus rounded y_i, and with linear distribution of $H(\rho, \theta)$ or updating by 1. Characteristic of these plots are errors decreasing with line length (the more points, the better the estimates) and an ordering in the accuracy of the methods, from best to worst, of least squares, Hough array smoothing and interpolating, centroid, and directly selecting the (first) maximum in the Hough array.

5.1 Special Cases

There are a few special cases. Figure 12, which was derived from rounded y_i, has an irregular angular dependence, with spikes near 45°, 63°, and 90°. Similar spikes were noticed in Gordon and Seering (1986, 1988). These spikes are at θ values corresponding to line slopes $m = \Delta y/\Delta x$, where Δx and Δy are in the ratio of 0:1, 1:1, and 2:1. Near these slopes the effect of rounding noisy coordinates can produce larger errors. When the quantizations $\Delta\rho$ and $\Delta\theta$ are very fine (as in Figure 12a), the counts are spread over many cells so that the signal-to-noise ratio is low. Combined with the rounding, this can cause the maximum to occur at an incorrect

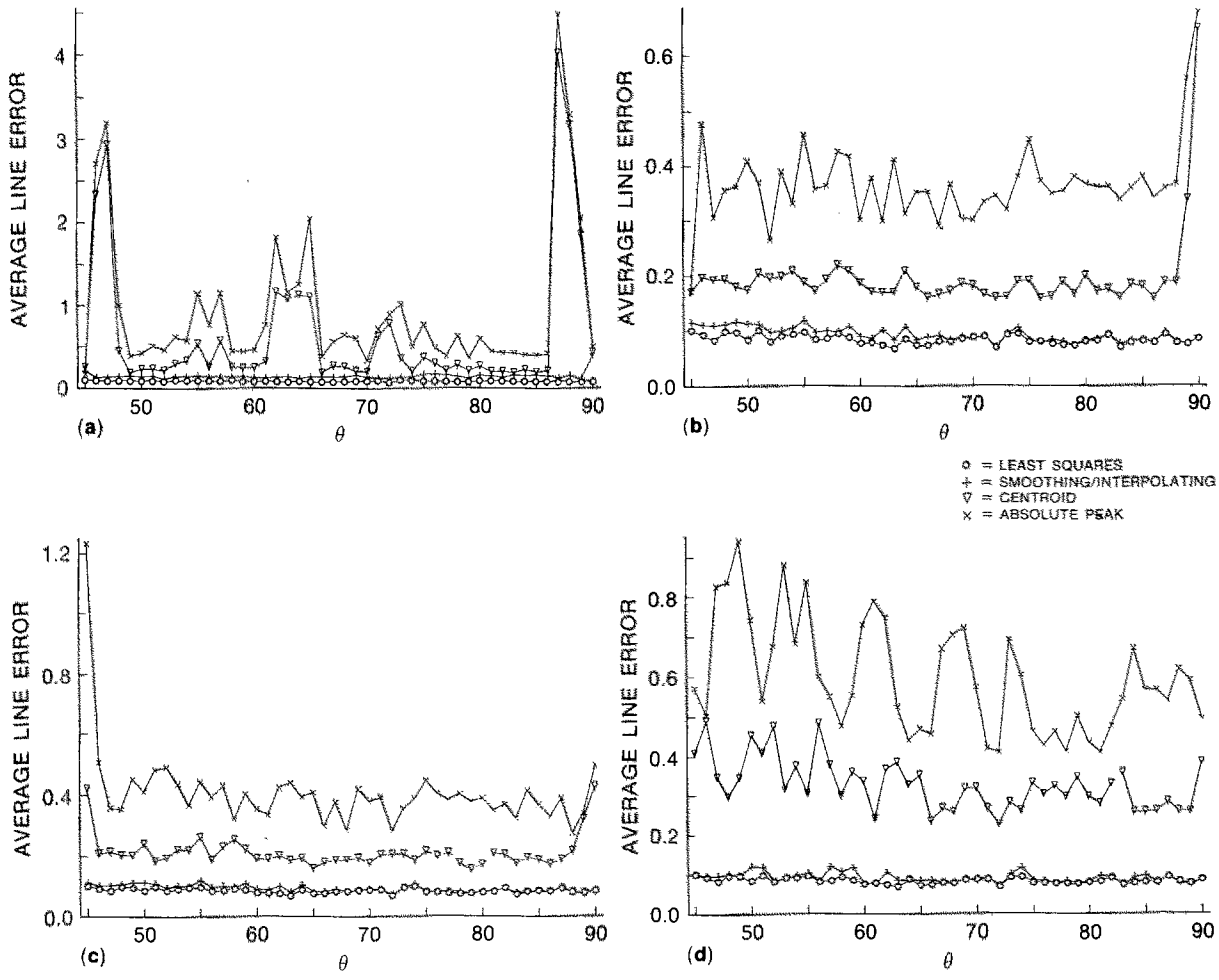

Figure 12. Average line errors for four quantizations of $\Delta\rho$ and $\Delta\theta$. Same as Figure 11 but rounded y_i. (a) $\Delta\rho = 0.1$, $\Delta\theta = 0.05$; (b) $\Delta\rho = 0.4$, $\Delta\theta = 0.25$; (c) $\Delta\rho = 0.5$, $\Delta\theta = 0.5$; (d) $\Delta\rho = 1$, $\Delta\theta = 1$.

location, or multiple maxima can occur in which case we take the first one. In either case the result is large errors. An example is shown in Figure 16, where the rounding causes a (relatively) large error in a computed ρ value at certain θ's, and this causes a large error in the final interpolated ρ value. The smoothing methods and least squares fitting are much less susceptible to this problem. In other respects (e.g., no trends) the plots for integer y_i are similar to those for exact y_i in Figure 11.

Another special case is shown in Figure 17. This is the case of no noise and shows the inherent error in the methods, such as that due to approximations in the linear distribution and interpolation schemes. Plots show the average line error for both exact and rounded y_i. These were computed from one trial (since there was no need for averaging) with $\Delta\rho = 0.1$, $\Delta\theta = 0.05$, and $H(\rho, \theta)$ incremented by linear distribution. The errors are extremely small, often less than 0.04 for all methods at the fine quantizations used. The small but noticeable negative bias for short lines is due to picking the first maximum when multiple maximums occur.

Figure 18 shows the effect of removing the linear distribution during the accumulation of $H(\rho, \theta)$. Compare this with the plots in Figure 15. These are for identical runs, but for Figure 18 we did not use the linear distribution but incremented the Hough array with 1. The linear distribution improves the average line error, although it is quite small regardless. This was also true for plots with y_i rounded to integer and for the different values of $\Delta\rho$ and $\Delta\theta$.

Finally, we show a plot with non-zero-mean noise. In all plots thus far the noise (if present) has been zero mean, and the least squares method is the most accurate. As a reminder of the utility of the

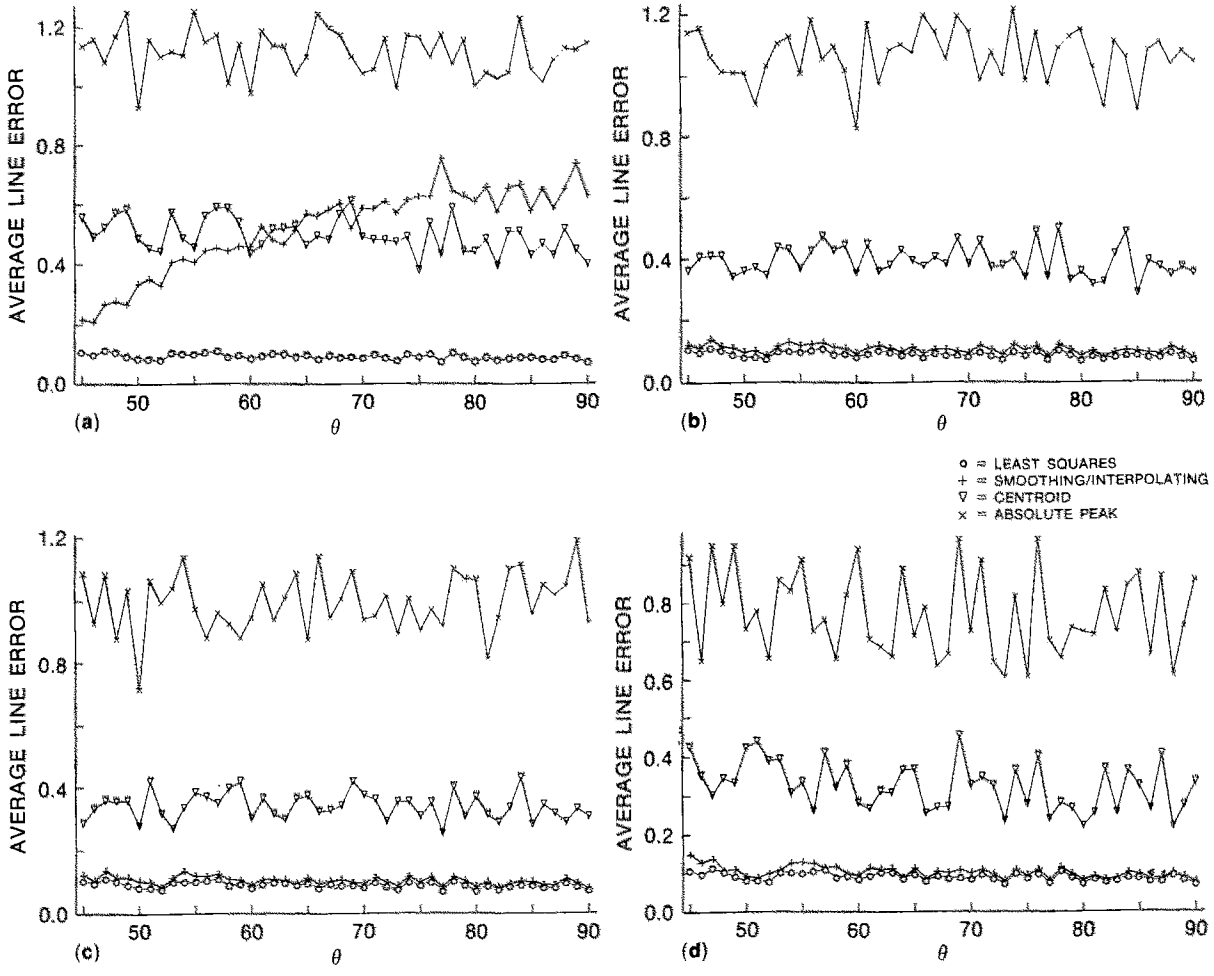

Figure 13. Average line errors for four quantizations of $\Delta\rho$ and $\Delta\theta$. Same as Figure 11 but zero-mean uniform noise of spread ± 2. (a) $\Delta\rho = 0.1$, $\Delta\theta = 0.05$; (b) $\Delta\rho = 0.4$, $\Delta\theta = 0.25$; (c) $\Delta\rho = 0.5$, $\Delta\theta = 0.5$; (d) $\Delta\rho = 1$ and $\Delta\theta = 1$.

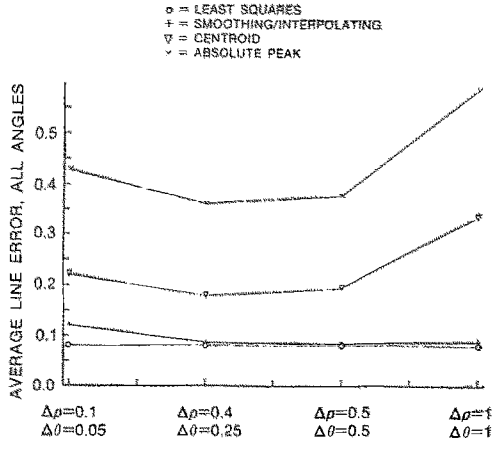

Figure 14. Summary plot of the effect of varying $\Delta\rho$ and $\Delta\theta$.

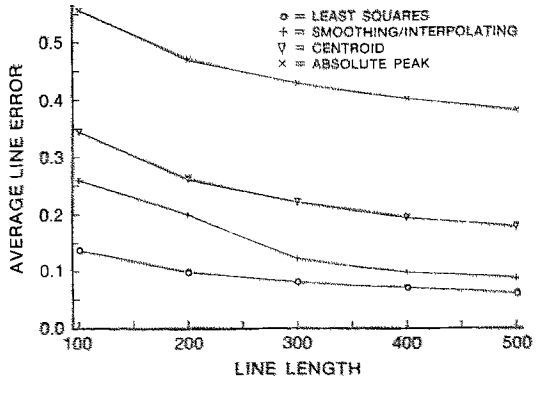

Figure 15. Average line error as a function of the line length. $\Delta\rho = 0.5$, $\Delta\theta = 0.5$, update of $H(\rho, \theta)$ by linear distribution, $\sigma_{\text{Noise}} = 1.0$, full precision y_i, $\rho = 15.25$.

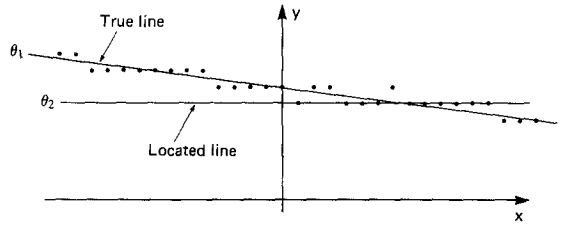

Figure 16. Example of large errors for rounded y_i and line with slope near 0. Final ρ and θ values are computed by interpolating using poor values from line at θ_2.

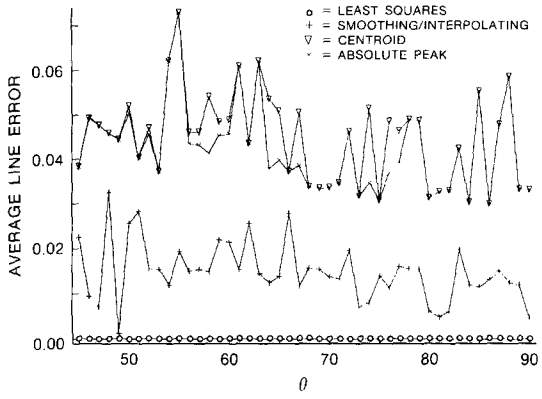

Figure 17. Errors inherent in the algorithm. Results are from runs with no (extra) noise added to y_i. $\Delta\rho = 0.1$, $\Delta\theta = 0.05$, update of $H(\rho, \theta)$ by liner distribution, $\sigma_{\text{Noise}} = 0$, $\rho = 15.25$.

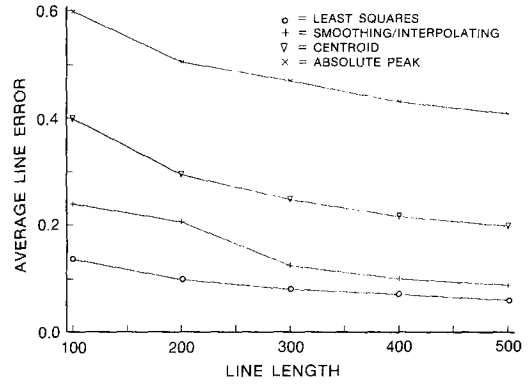

Figure 18. Removing the linear distribution. All other parameters are as those for Figure 15.

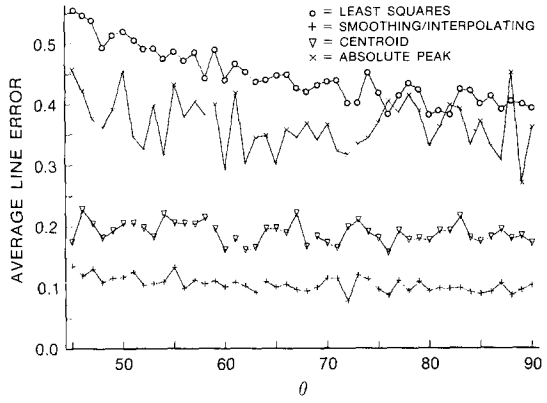

Figure 19. Results when "mousebite" noise is added. All other parameters are as those for Figure 15.

Hough transform and its robustness to many types of noise, we show an example in which two "mousebites" were added to the input points. The runs were identical to those shown in Figure 11, but for each line (of length 300) the y value in two sections of length 10 was offset by +4. The results are shown in Figure 19. The Hough methods are essentially unaffected, whereas the least squares method is not.

6 Experimental Results

The accuracy of any high precision measurement system will depend on all components: mechanical, optical, electronic, and algorithmic. We wanted to avoid such issues as optical and electronic distortion, shading problems, and synchronization between camera and digitizer, so we relied mostly on the simulations. Our experimental results serve to illustrate and compare the methods, rather than to give absolute accuracy figures.

The images we used were obtained using an Olympus microscope with VSP model SC505 CCD camera, attached to a Matrox MVP-AT vision board installed in an IBM PC AT. We neglected the slight shading problems in the images. The camera/electronics aspect ratio was carefully calibrated by repeated measurements of several widths on a high precision target. The x and y pixel dimensions were measured separately to give 0.8367 microns per pixel in x and 0.8734 in y. When digitizing, synchronization was provided by the camera. In order to avoid blurring and jitter errors due to possible synchronization problems between the camera and digitizer, (1) the images were not frame averaged and (2) we performed relative measurements such that we use distance or angle between two lines in the same frame. It was assumed that the objects are planar, so that no 3-dimensional calibration was done.

Two targets of chrome-on-glass patterns were used. For ρ measurements the target was a pattern of stripes of thin lines at 100 micron spacing. For θ

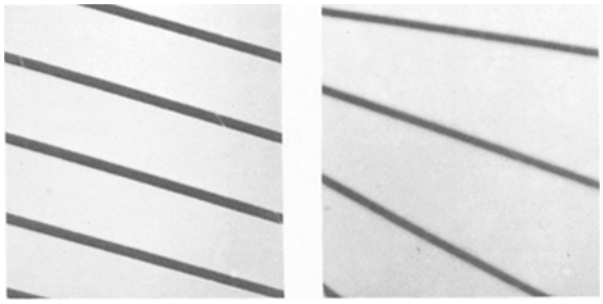

Figure 20. Targets used in experiments. (a): ρ target. (b): θ target.

the target was a microscope eyepiece with calibrated angular lines at 10° intervals. For the θ target we did not have exact manufacturer's specifications. Figure 20 shows the targets.

Once digitized, the images were processed as follows:

1. Compute the vertical gradient with a [−1, 0, 1] operator.
2. Take only the positive gradient and manually threshold at approximately 50 percent of the peak gradient. In this way we only extracted leading edges of the target.
3. Extract the (x_i, y_i) coordinates of the thresholded edgels, calibrate them by multiplying by x and y calibration constants (microns per pixel in the x and y directions), and input them to the programs used in the simulations.

For ρ measurements we took images of the target at angles θ of approximately 90°, 75°, 64°, and 45°. At each angle an image was taken, the target slightly shifted, and another image taken. This was repeated again to give three images at each of the four angles, or 12 measurements in all. The images were thresholded, giving lines that had a length of about 270. They were from 2 to 4 pixels wide, and so we used $b = 1.5$. We chose $\Delta\theta = 0.25$, and from Eq. (5) we chose $\Delta\rho = 0.5$ to give a maximum spread at the peak of 6–12 buckets. This is below the maximum $\Delta\rho$ given by Eq. (8), so it is acceptable.

The distance d between lines was taken as the main measurement and its mean $\bar{d}$ and variance from 100 microns ($\sigma^2(d - 100)$) were computed for each of the four methods—least squares fit, smoothing and interpolation of $H(\rho, \theta)$, centroid, and simple maximum in Hough array—for each of the 12 measurements. To avoid the dependence of ρ on the coordinate origin and θ, we computed the distance measurement as the distance along the line normal to the average angle of two lines and that passes through the centroid of the set of (x_i, y_i) of both lines. Because two edges are measured, the variance of the ρ estimate is approximately half the distance variance. The results (in microns) are given in Table 1.

Table 1. The 12 mesurements d for the distance between two (nominally) parallel edges 100 microns apart[a]

	L	G	C	A
1	100.27	100.31	99.97	99.5
2	100.31	100.32	100.50	99.5
3	100.29	100.29	99.98	100.50
4	99.94	99.93	99.87	99.53
5	100.01	100.00	100.16	99.55
6	99.92	99.92	99.79	99.45
7	100.08	100.08	100.12	99.89
8	100.11	100.11	99.98	99.92
9	99.89	100.01	100.11	99.84
10	99.85	100.08	99.97	100.53
11	100.14	100.01	99.99	100.56
12	100.01	100.17	100.20	100.55
Mean	0.068	0.103	0.053	−0.057
σ	0.153	0.136	0.176	0.445

[a] The columns show measurements from least squares (L), smoothing and interpolation (G), centroid (C), and absolute Hough peak (A). At the bottom are the mean and standard deviation for $d - 100$. Values are in microns.

All methods (except simple maximum) provide approximately the same accuracy. The least squares fit was probably influenced by the nonsymmetrical noise in the thick lines after thresholding the gradient. The accuracy is within the range predicted by the simulation.

For the θ measurements we took five images of the angular target, Figure 20b. Each image was taken after the target had been slightly moved (translated and rotated). We obtained seven measurements since two images contained two usable measurements. The main measurement was the angle between two lines, and we assumed that the least squares fit gave the best estimate since we did not have the exact target specifications. Lines were approximately 300 pixels long and 6 to 7 pixels wide. We used $\Delta\theta = 0.25$, $\Delta\rho = 0.5$, and $b = 3$, corresponding to the line half-width. We compared results from the three Hough methods by computing their mean and standard deviation with respect to the least squares values. These are shown in Table 2. The standard deviation again has to be divided by 2, which brings this measurement within the range of results predicted by simulation.

Table 2. The seven mesurements of angle[a]

	L	G	C	A
1	10.261	10.273	10.226	10.5
2	10.186	10.189	10.184	10.0
3	10.354	10.359	10.320	11.0
4	10.451	10.453	10.443	10.25
5	10.195	10.192	10.163	11.5
6	10.121	10.166	10.158	9.5
7	10.142	10.164	10.193	9.75
Mean		0.0123	−0.003	0.113
σ		0.0153	0.0326	0.621

[a] The columns show measurements from least squares (L), smoothing and interpolation (G), centroid (C), and absolute Hough peak (A). At the bottom are the mean and standard deviation of the difference of the measurement with respect to the least squares measurement. Values are in microns.

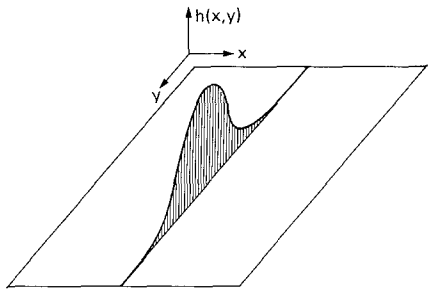

Figure 21. The spatial filter corresponding to the Gaussian projection domain filter.

7 Spatial Filters Equivalent to the $H(\rho, \theta)$ Filters

The smoothing portion of the method we have presented essentially describes a filter that is applied to the columns of Hough space $H(\rho, \theta)$ as a preprocessing step before finding the peak. In many cases an operation in projection space has a corresponding filter in image space (Deans 1983; Hinkle et al. 1987), and we can ask the question: Is there an equivalent spatial domain filter that can be applied directly to the image data to produce the same result that we get by applying the projection space filter to $H(\rho, \theta)$. The answer is yes, and the spatial filter is shown in Figure 21. A derivation is given in the appendix. The filter is surprising in its simplicity, but, in fact, it is what would be expected from the y-axis-only Gaussian filters we are applying in projection space.

8 Conclusions

We have described several sources of error in the standard Hough transform and have proposed practical methods to overcome them. The methods yield very high accuracy estimates of straight-line parameters, significantly higher than generally recognized as possible from the Hough transform. The error due to rounding ρ during accumulation of $H(\rho, \theta)$ is negligible in most cases, but errors due to peak spreading and the quantization of ρ and θ are not. For the main error, due to the spreading of the peak, we presented a method that preprocesses the Hough array to greatly reduce this error. For the error due to the quantization of ρ and θ in $H(\rho, \theta)$ we described an interpolation that results in final accuracies that can be much better than $\Delta\rho$ and $\Delta\theta$. We also suggested several guidelines for selecting $\Delta\rho$ and $\Delta\theta$.

The methods are based on the geometry of the spreading of the peak in $H(\rho, \theta)$ and on straightforward interpolation schemes. We do not claim that the methods are optimal, but we demonstrated their validity in extensive simulations and in selected experiments on real targets. Their use significantly improves the accuracy of the results as compared with the standard Hough transform, making them very close in accuracy to least squares even for the case of Gaussian zero-mean noise, where least squares is optimal. Compared with the standard method of taking the absolute peak in the Hough array, the new methods give considerable improvement, reducing the error in the computed line, as we measured it, by factors from 3 to 10.

Least squares and robust least squares are highly accurate and suitable for many cases. However, in particular cases of background or other non-zero-mean noise, the modified Hough transform and robust least squares performed much better than least squares whereas in cases of multiple lines the Hough transform accurately identified one of the lines while even robust least squares fit an incorrect "average" line. In general, including situations with multiple lines and various types of noise, our experience suggests that some combination of Hough transform and least squares or robust least squares is the best choice—for example, an initial Hough transform, including peak smoothing, to identify the relevant points, followed by least squares on those points that contribute to the peak.

Acknowledgment. We are grateful to B. Dom, M. Flickner, L. Barbosa, and J. Sanz for helpful discussions related to this work.

Appendix: Derivation of Equivalent Spatial Filters

To begin, we state several standard results. The first are summarized from Rosenfeld and Kak (1982, pp. 365–370). Let $F(u, v)$ be the 2-D Fourier transform of $f(x, y)$ expressed in rectangular coordinates. Then by definition of the inverse Fourier transform

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v) e^{j2\pi(ux + vy)}\, du\, dv$$

If $F(u, v)$ is expressed in polar coordinates as $F(\omega, \theta)$, the inverse transform is

$$\begin{aligned} f(x, y) &= \int_0^{2\pi} \int_0^{+\infty} F(\omega, \theta)\omega e^{j2\pi\omega(x\cos\theta + y\sin\theta)}\, d\omega\, d\theta \\ &= \int_0^{\pi} \int_{-\infty}^{+\infty} F(\omega, \theta)|\omega| e^{j2\pi\omega(x\cos\theta + y\sin\theta)}\, d\omega\, d\theta \end{aligned} \tag{13}$$

The Slice Theorem associated with the Radon transform relates the Fourier transform of a 1-D projection of $f(x, y)$ to a slice through the 2-D Fourier transform. In particular, if $S_\theta(\omega)$ is the 1-D Fourier transform of the projection $P_\theta(\rho)$ of $f(x, y)$ at angle θ, $F(u, v)$ the 2-D Fourier transform, and $F(\omega, \theta)$ the slice of $F(u, v)$ at angle θ, then the Slice Theorem states that

$$S_\theta(\omega) = F(\omega, \theta) \tag{14}$$

Combining the Slice Theorem with the polar form of the inverse Fourier transform, Eq. (13), and doing some manipulations to change the limits of integration, we obtain

$$f(x, y) = \int_0^{\pi} \int_{-\infty}^{+\infty} S_\theta(\omega)|\omega| e^{j2\pi\omega(x\cos\theta + y\sin\theta)}\, d\omega\, d\theta \tag{15}$$

Two additional equations we will use are taken from Bracewell (1978). These are the formula for a particular integral yielding the delta function:

$$\int_{-\infty}^{+\infty} e^{-j2\pi xs}\, ds = \delta(x) \tag{16}$$

and the (inverse) Fourier transform of a Gaussian with a particular scaling:

$$\int_{-\infty}^{+\infty} e^{-\pi(\sqrt{2\pi}\sigma t)^2} e^{-j2\pi xt}\, dt = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \tag{17}$$

Now we are ready to derive the spatial domain filter corresponding to the projection domain filter we defined in the body of the paper. We will denote the projection domain filter by g. What we are looking for is a filter $h(x, y)$ such that

$$\tilde{f}(x, y) = h(x, y) * f(x, y)$$

where $f(x, y)$ is our original image, $\tilde{f}(x, y)$ is the image corresponding to the projections filtered by g, and $*$ represents convolution. In other words, for a given filter g applied to the (continuous) Hough space $H(\rho, \theta)$, $\tilde{f}(x, y)$ is given by

$$\tilde{f}(x, y) = \int_0^{\pi} \int_{-\infty}^{+\infty} \tilde{S}_\theta(\omega)|\omega| e^{j2\pi\omega(x\cos\theta + y\sin\theta)}\, d\omega\, d\theta \tag{18}$$

where $\tilde{S}_\theta(\omega)$ is the Fourier transform of the θ column of $H(\rho, \theta)$, or equivalently the projection $P_\theta(\rho)$ of $f(x, y)$ at angle θ, filtered by g. That is,

$$\tilde{S}_\theta(\omega) = F(g_\theta(\rho) * P_\theta(\rho))$$

where F is the Fourier transform operator. By the convolution theorem for the Fourier transform,

$$\tilde{S}_\theta(\omega) = F(g_\theta(\rho)) \times F(P_\theta(\rho)) = G_\theta(\omega) \times S_\theta(\omega)$$

where $G_\theta(\omega)$ is the Fourier transform of $g_\theta(\rho)$. For the Gaussian filters we used

$$g_\theta(\rho) = \frac{1}{\sqrt{2\pi}\sigma \sin\theta} e^{-\rho^2/(2(\sigma\sin\theta)^2)}$$

so that from (17)

$$G_\theta(\omega) = e^{-(2\pi\sigma\sin\theta\omega)^2/2} = e^{-\pi((\sqrt{2\pi}\sigma\sin\theta)\omega)^2}$$

Then we can write Eq. (18) as

$$\tilde{f}(x, y) = \int_0^{\pi} \int_{-\infty}^{+\infty} G_\theta(\omega) S_\theta(\omega)|\omega| e^{j2\pi\omega(x\cos\theta + y\sin\theta)}\, d\omega\, d\theta$$

By the Slice Theorem

$$\tilde{f}(x, y) = \int_0^{\pi} \int_{-\infty}^{+\infty} G_\theta(\omega) F(\omega, \theta)|\omega| e^{j2\pi\omega(x\cos\theta + y\sin\theta)}\, d\omega\, d\theta$$

Changing the limits of integration (which we can do because $G_\theta(\omega) = G_\theta(-\omega)$), we obtain

$$\tilde{f}(x, y) = \int_0^{2\pi} \int_0^{+\infty} G_\theta(\omega) F(\omega, \theta)\omega e^{j2\pi\omega(x\cos\theta + y\sin\theta)}\, d\omega\, d\theta$$

which is the inverse Fourier transform of $[G\theta(\omega)F(\omega, \theta)]$, or

$$\tilde{f}(x, y) = F^{-1}[G_\theta(\omega)F(\omega, \theta)]$$

Again, by the Convolution Theorem

$$\tilde{f}(x, y) = F^{-1}(G_\theta(\omega)) * F^{-1}(F(\omega, \theta)) = F^{-1}(G_\theta(\omega)) * f(x, y)$$

This says that $\tilde{f}(x, y)$ is given $\tilde{f}(x, y)$ convolved with $F^{-1}(G_\theta(\omega))$, or that $F^{-1}(G_\theta(\omega))$ is the spatial filter we are trying to find. The inverse polar Fourier transform is

$$F^{-1}(G_\theta(\omega)) = \int_0^{2\pi} \int_0^{\infty} \omega e^{-\pi((\sqrt{2\pi}\sin\theta)\omega)^2} e^{j2\pi\omega(x\cos\theta + y\sin\theta)} d\omega\, d\theta$$

Making the change of variables $u = \omega \cos\theta$, $v = \omega \sin\theta$, we get

$$F^{-1}(G_\theta(\omega)) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-\pi(\sqrt{2\pi}\sigma v)^2} e^{j2\pi(xu+yv)}\, du\, dv$$

$$= \int_{-\infty}^{+\infty} e^{-\pi(\sqrt{2\pi}\sigma v)^2} e^{j2\pi yv}\, dv \int_{-\infty}^{+\infty} e^{j2\pi xu}\, du$$

Using Eq. (17) for the first term and Eq. (16) for the second, this reduces to

$$F^{-1}(G_\theta(\omega)) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-y^2/2\sigma^2)} \delta(x)$$

This is the spatial filter $h(x, y)$ we want. It is shown in Figure 21, and clearly corresponds to the y-axis Gaussian filters we are applying in projection space.

References

Ambs P, Lee SH, Tian Q. Fainman Y (1986) Optical implementation of the Hough transform by a matrix of holograms. Applied Optics 25:4039–4045

Ballard DH (1981) Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition 13:111–122

Ballard DH (1987) Interpolation coding: A representation for numbers in neural models. Biological Cybernetics 57:389–402

Ballard DH, Brown CM (1982) Computer vision, Prentice-Hall, Englewood Cliffs, NJ, pp 123–130

Baringer WB, Richards BC, Brodersen RW, Sanz JLC, Petkovic D (1987) A VLSI implementation of PPPE for real-time image processing in Random space — Work in progress. Proceedings of IEEE Workshop on Computer Architectures for Pattern Analysis and Machine Intelligence, Seattle, WA

Bracewell RN (1978) The Fourier transform and its applications. McGraw-Hill New York

Brown CM (1983) Inherent bias and noise in the Hough transform. IEEE Pattern Analysis and Machine Intelligence PAMI-5 (5):493–505

Canny J (1986) A computational approach to edge detection. IEEE Pattern Analysis and Machine Intelligence PAMI-8 (6):679–698

Cohen M, Toussaint GT (1977) On the detection of structures in noisy pictures. Pattern Recognition 9:95–98

Davies ER (1987) A new framework for analysing the properties of the generalized Hough transform. Pattern Recognition Letters 6:1–7

Deans SR (1981) Hough transform from the Randon transform. IEEE Pattern Analysis and Machine Intelligence PAMI-3(2):185–188

Deans SR (1983) The Radon transform and some of its applications. John Wiley and Sons, New York

Duda RO, Hart PE (1972) Use of the Hough transform to detect lines and curves in pictures. Communication of the ACM 15:11–15

Dyer CR (1983) Gauge inspection using Hough transform. IEEE Pattern Analysis Machine Intelligence PAMI-5(6):621–623

Eichmann G, Dong BZ (1983) Coherent optical production of the Hough transform, Applied Optics 22:830–834

Galkowski JT, Galkowski PJ (1986) On the importance of signal-to-noise measures in machine vision: A case study of the Hough transform. IBM Federal Systems Division Technical Report 86-L57-003, Owego, NY

Gindi GR, Gmitro AF (1984) Optical feature extraction via the Radon transform. Optical Engineering 23:499–506

Gordon SJ, Seering WP (1986) Accuracy issues in measuring quantized images of straight-line features. IEEE International Conference on Robotics and Automation, San Francisco, CA, pp 931–936

Gordon SJ, Seering WP (1988) Real-time part position sensing. IEEE Pattern Analysis and Machine Intelligence PAMI-10(3):373–386

GRAFSTAT User's Guide (1986), available from IBM

Hanahara K, Maruyama T, Uchiyama T (1988) A real time processor for the Hough transform. IEEE Pattern Analysis and Machine Intelligence PAMI-10(1):121–125

Hinkle E, Sanz JLC, Jain AK, Petkovic D (1987) PPPE: New life for projection-based image processing. Journal of Parallel and Distributed Computing (4):45–78

Holland PW, Welsch RE (1977) Robust regression using iteratively reweighted least squares. Communications Statistics-Theoretical Methods A6(9):813–827

Hough PVC (1962) Method and means for recognizing complex patterns. U.S. Patent 3069654

Huber PJ (1981) Robust statistics. John Wiley and Sons, New York

Hunt DJ, Nolte LW, Ruedger WH (1988) Performance of the Hough transform and its relationship to statistical signal detection theory. Computer Vision, Graphics, and Image Processing 43(2):221–238

Illingworth J, Kittler J (1987) The adaptive Hough transform. IEEE Pattern Analysis and Machine Intelligence PAMI-9(5):690–698

Illingworth J, Kittler J. (1988) A survey of the Hough

transform. Computer Vision, Graphics, and Image Processing 44:87–116

Kimme C, Ballard DH, Sklansky J (1975) Finding circles by an array of accumulators. Communications of the ACM 18:120–122

Kiryati N, Bruckstein AM (1988) Antialiasing the Hough transform. EE Publication 697, Department of Electrical Engineering, Technion, Haifa, Israel

Leavers VF, Boyce JF (1986) An implementation of the Hough transform using a linear array processor in conjunction with a PDP/11. National Physical Laboratory, Teddington, Middlesex, England, NPL Report DITC 74/86

Li CC, Mancuso JF, Shu DB, Sun YN, Roth LD (1983) A preliminary study of automated inspection of VLSI resist patterns. Proceedings of the IEEE International Conference on Robotics and Automation, pp 474–480

Li H, Lavin MA, LeMaster RJ (1986) Fast Hough transform: A hierarchical approach. Computer Vision, Graphics, and Image Processing 36:139–161

Maitre H (1986) Contribution to the prediction of performance of the Hough transform. IEEE Patten Analysis and Machine Intelligence PAMI-8(5)669–674

Niblack W, Petkovic D (1986) On improving the accuracy of the Hough transform, theory, simulation, and experiments. Proceedings of Computer Vision and Pattern Recognition, Ann Arbor, Michigan, June, pp 574–579

Rhodes FM, Dituri JJ, Chapman GH, Emerson BE, Soares AM, Raffel JI (1988) A monolithic Hough transform processor based on restructurable VLSI. IEEE Pattern Analysis and Machine Intelligence PAMI-10(1):106–110

Rosenfeld A, Kak AC (1982) Digital picture processing. Second Edition. Volume 1. Academic Press, New York

Shapiro SD (1978) Generalization of the Hough transform for curve detection in noisy digital images. Fourth International Joint Conference on Pattern Recognition, Kyoto, Japan

Shapiro SD (1978) Properties of transforms for the detection of curves in noisy pictures. Computer Graphics and Image Processing 8:219–236

Shapiro SD, Iannino A (1979) Geometric constructions for predicting Hough transform performance. IEEE Pattern Analysis and Machine Intelligence PAMI-1(3):310–317

Sheinvald J, Dom B, Niblack W (1989) Multiple curve detection using the MDL principle and the Hough transform. IBM Almaden Research Center

Sklansky J (1978) On the Hough technique for curve detection. IEEE Transactions on Computers C-27, (10):923–926

Srihari SN, Govindaraju V, (1989) Analysis of textual images using the Hough transform. Machine Vision and Applications 2(3):141–153

Steier WH, Shori RK (1986) Optical Hough transform. Applied Optics 25:2734–2738

Thrift PR, Dunn SM (1983) Approximating point set images by line segments using a variation of the Hough transform. Computer Vision, Graphics and Image Processing 21:383–394

Van Veen TM, Groen FCA (1981) Discretization errors in the Hough transform. Pattern Recognition 14:137–145

Young RA (1986) Locating parts with subpixel accuracies. SPIE, Boston, MA